EXHIBIT J

TO DECLARATION OF S. MERRILL WEISS IN SUPPORT OF PLAINTIFF ACACIA MEDIA TECHNOLOGIES CORPORATION'S MEMORANDUM OF POINTS AND AUTHORITIES IN OPPOSITION TO ROUND 3 DEFENDANTS' MOTION FOR SUMMARY JUDGMENT OF INVALIDITY UNDER 35 U.S.C. § 112 OF THE '992, '863, AND '702 PATENTS; AND SATELLITE DEFENDANTS' MOTION FOR SUMMARY JUDGMENT OF INVALIDITY OF THE '992, '863, AND '720 PATENTS

Hovsepian v. Apple, Inc. Doc. 325 Att. 9

Dockets.Justia.com

Fixed and Adaptive Predictors for Hybrid Predictive/Transform Coding

STAFFAN ERICSSON

Abstract—Hybrid predictive/transform coding is studied. The usual formulation is to first apply a unitary transform and then code the transform coefficients with independent DPCM coders, i.e., the prediction is performed in the transform domain. This structure is compared to spatial domain prediction, where a difference signal is formed in the spatial domain and then coded by a transform coder. A linear spatial domain predictor which minimizes the mean square prediction error also minimizes the mean square of each transform coefficient. The two structures are equivalent if the transform domain prediction scheme is extended to a more general predictor. Hence, the structure that gives the easiest implementation can be chosen. The spatial domain structure is preferred for motion compensation and for line interlaced video signals. Interframe hybrid coding experiments are performed on interlaced videophone scenes using an adaptive transform coder. Motion compensation gives a rate reduction of 25-35 percent compared to frame difference prediction with the same mean square error. The subjective advantage is even greater, since the "dirty window" effect is not present with motion compensation. It is important to perform the motion estimation with fractional pel accuracy. Field coding with a switched predictor using previous field in moving areas is an interesting alternative to frame coding with frame difference prediction.

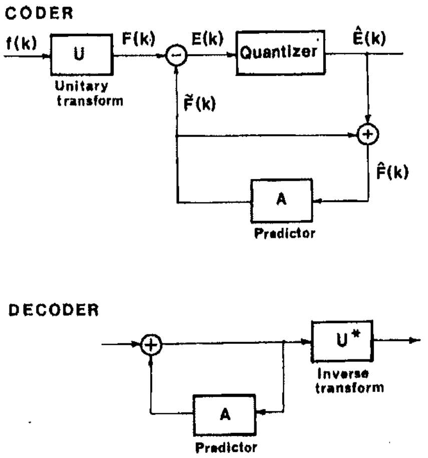


Fig. 1. Structure I: hybrid coder with transform domain prediction.

I. Introduction

PREDICTIVE coding and transform coding are the two most popular techniques for picture coding; see, e.g., the review by Netravali and Limb [1]. Transform coding often performs better at low bit rates, but predictive coding offers easier implementation [2]. In hybrid coding the two techniques are combined. For intraframe coding a transform is applied along the line, and each transform coefficient is processed by a separate DPCM loop to take care of the statistical dependence in the vertical direction [3]–[6]. In interframe hybrid coding, a two-dimensional transform is applied intraframe, and the transform coefficients are DPCM encoded in the temporal dimension [7]–[16].

Hybrid coding is a compromise which usually gives performance and complexity between transform and predictive coding; however, for interframe coding, Roese *et al.* [9] reported better results with adaptive hybrid coding than with three-dimensional transform coding. Moreover, hybrid interframe coding allows the inclusion of motion compensation techniques, which can give dramatic improvements in bit rate reduction for many scenes. There is no technique known to the author by which motion compensation can be applied to three-dimensional transform coding.

A block diagram of a hybrid coding system is shown in Fig. 1. The source is a sequence of N-dimensional vectors $\{f(k)\}$. Each vector $f(k)$ is transformed by a unitary transformation U into a new vector $F(k)$,

$$F(k) = Uf(k).$$

A prediction $\tilde{F}(k)$ is formed from the previous reconstructed transformed vector $\hat{F}(k-1)$, which is also available to the receiver. In the original hybrid coding scheme by Habibi [3], the transform coefficients are coded by independent DPCM loops, one for each coefficient. In Fig. 1 all operations are performed on vectors, i.e., the prediction of each transform coefficient could be a linear combination of several reconstructed transform coefficients. An example will be given in Section II.

The transform domain prediction error $E(k)$,

$$E(k) = F(k) - \tilde{F}(k)$$

is quantized by the quantizer Q. In this paper we assume a componentwise quantization,

$$\hat{E}_i(k) = Q_i(E_i(k)), \qquad i = 1, 2, \cdots, N.$$

The above scheme is called *Structure I* in the sequel.

A different hybrid coding structure is outlined in Fig. 2. The order between transformation and prediction is reversed, so that the predictor works in the space domain. The space domain prediction error $e(k)$ is then transformed and quantized. We call the scheme *Structure II*. In this scheme an inverse transformation U^* must be performed in the feedback loop, since the predictor operates in the space domain.

In Structure II we can directly apply techniques from predictive coding to make the predictor more efficient. Forchheimer and Ericson [17], [18] apply the structure to intraframe coding. They use a least mean square predictor with five elements in the previous line and a 16-point discrete cosine transform (DCT) [18]. The five-element predictor gave a favorable spectrum shape, with less variance in the high frequency coefficients compared to a predictor using one

Paper approved by the Editor for Signal Processing and Communication Electronics of the IEEE Communications Society. Manuscript received December 4, 1984; revised July 15, 1985. This paper was presented in part at the Picture Coding Symposium, Davis, CA, March 1983. This work was performed in partial fulfillment of the requirements of the Ph.D. degree from the Royal Institute of Technology, Stockholm, Sweden, and was supported by the Swedish National Board for Technical Development under Contracts 80-3594 and 82-3401.

The author was with INRS-Telecommunications, Verdun, P.Q., Canada. He is now with PicTel Corporation, Peabody, MA 01960.

0090-6778/85/1200-1291$01.00 © 1985 IEEE

Exhibit J Page 194

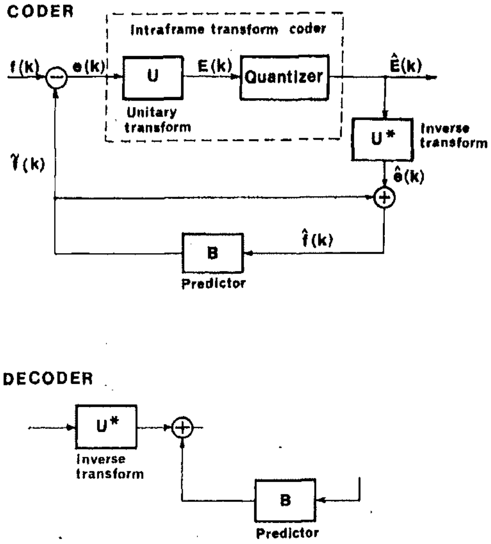


Fig. 2. Structure II: hybrid coder with space domain prediction.

element only. Wilson *et al.* [19], [20] used a similar scheme combined with anisotropic postfiltering. Side information about edge and line content was transmitted to control the anisotropic filter and an adaptive predictor.

Jain and Jain [12] used motion compensation in a hybrid interframe coder, and they suggested Structure II because the motion compensation algorithm worked in the space domain. Motion compensation is a kind of adaptive prediction where the prediction is taken as the elements from previous frame displaced according to a motion estimate. The interframe hybrid coder by Telese and Zarone [21] also used Structure II. The frame difference was formed, and the picture was segmented in moving and nonmoving areas. The frame differences in moving areas were transform coded with a variable transform block size to fit the shape of the moving area.

The aim of this study is to exploit the possibilities of efficient prediction in hybrid coding offered by Structure II, namely adaptive prediction (e.g., motion compensation) and predictors that utilize the previous field with a line interlaced signal.

In Section II we discuss space domain versus transform domain prediction in hybrid coding, and we show that Structures I and II are equivalent. In Section III it is shown that, under certain assumptions, a least mean square predictor is also optimum in a hybrid coder. Fixed and adaptive predictors, including motion compensation, are suggested for interframe hybrid coders and applied to typical videophone scenes in Section IV. Results for a complete adaptive hybrid coder are presented in Section V.

II. Space Domain Versus Transform Domain Prediction

In this section we will compare the two hybrid coding structures—transform domain and space domain prediction — outlined in Figs. 1 and 2, respectively.

The source is a sequence of N-dimensional vectors $\{f(k)\}$,

$$f(k)=(f_1(k)f_2(k)\cdots f_N(k))^T. \tag{1}$$

In an interframe 2-D transform/predictive hybrid coder, each vector consists of a frame with $N = N_1N_2$ picture elements (pels) and k indicates frame number. In intraframe hybrid coding—applying 1-D transform in the horizontal direction and DPCM in the vertical direction—each line is regarded as a vector, and k indicates line number.

A. Structure I—Transform Domain Prediction

In Structure I (Fig. 1) each vector is transformed by a unitary transformation,

$$F(k)=Uf(k). \tag{2}$$

Normally, the transformation is performed blockwise (e.g., on blocks of $K = 16 \times 16$ pels) to reduce the complexity and to take advantage of the varying statistics in different parts of the picture. In that case the transformation matrix U will have at most K nonzero elements in each row and column. With a suitable ordering of the elements in $f(k)$, the unitary transform is a block diagonal matrix,

$$U=\text{diag }(U_K \quad U_K \cdots U_K) \tag{3}$$

where U_K is the $K \times K$ transformation matrix for one block.

The transformed vector sequence $\{F(k)\}$ is coded by DPCM, i.e., a prediction $\tilde{F}(k)$ is generated from previous transformed vectors $\hat{F}(k-1), \hat{F}(k-2), \cdots$, which are also available to the receiver. Normally the predictor utilizes only one previous vector. A time-varying linear predictor is defined by

$$\tilde{F}(k)=A(k)\hat{F}(k-1). \tag{4}$$

The transform domain prediction error $E(k)$

$$E(k)=F(k)-\tilde{F}(k) \tag{5}$$

is quantized into $\hat{E}(k)$ and transmitted,

$$\hat{E}(k)=Q(E(k)). \tag{6}$$

The receiver reconstructs the transformed vector

$$\hat{F}(k)=\tilde{F}(k)+\hat{E}(k) \tag{7}$$

and performs an inverse transformation,

$$\hat{f}(k)=U^*\hat{F}(k). \tag{8}$$

Habibi [3], [4] utilizes independent DPCM coders with fixed prediction for the sequences of transform coefficients. The predictor takes the form

$$\tilde{F}(k)=A\hat{F}(k-1) \tag{9}$$

with A a diagonal matrix,

$$A=\text{diag }(a_1 \quad a_2 \cdots a_N). \tag{10}$$

The same form is used for interframe coding in [7]-[10], [14].

A nondiagonal matrix has been used by Chiariglione *et al.* [6] in intraframe coding of PAL composite color TV signals. They applied a 12-point discrete Fourier transform along the line. DPCM was used to code the three lowest order coefficients, and coefficients 10 and 11, which contain chrominance information. The prediction coefficient matrix A_{12}

$$A_{12}=\begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&t_{11}&t_{12}&0\\ 0&0&0&0&0&0&0&0&0&t_{21}&t_{22}&0\\ 0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix} \tag{11}$$

Exhibit J Page 195

was used for each transform block. The chrominance predictor T

$$T=\begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix} \tag{12}$$

performs a rotation of the PAL coded chrominance components to compensate for the change in color subcarrier phase between successive lines, i.e., the predictor performs a signal processing function in the frequency domain.

In several papers a fixed predictor coefficient α is used for all transform coefficients [11], [12], [15], [16]:

$$A=\alpha I \tag{13}$$

where I is the identity matrix and α is slightly less or equal to 1. An adaptive predictor was used by May [13] in interframe coding of slow scan television; α was selected and transmitted for each 16×16 transform block.

B. Structure II—Space Domain Prediction

In Structure II the prediction is made before the transformation; see Fig. 2. The prediction $\tilde{f}(k)$ is formed in the space domain using the previously reconstructed vector $\hat{f}(k-1)$. We assume a time-varying linear predictor,

$$\tilde{f}(k)=B(k)\hat{f}(k-1). \tag{14}$$

The space domain prediction error $e(k)$

$$e(k)=f(k)-\tilde{f}(k) \tag{15}$$

is transform coded, i.e., transformed, quantized, and transmitted.

In the Appendix it is shown that Structure II is equivalent to Structure I if we keep the same quantizer and choose the prediction coefficient matrix $B(k)$ to be

$$B(k)=U^*A(k)U \tag{16}$$

where $A(k)$ is the prediction coefficient matrix in Structure I according to (4). Equation (16) relates a space domain predictor and a transform domain predictor. An equivalent expression is

$$A(k)=UB(k)U^*. \tag{17}$$

Hence, Structures I and II are equivalent if the prediction matrices $A(k)$ and $B(k)$ are allowed to be nondiagonal. Note, however, that the matrix $A(k)$ may be considerably more complicated than the diagonal matrix in Habibi's scheme.

For the special case $A=\alpha I$ we obtain

$$B=A=\alpha I. \tag{18}$$

In this case Structure I is more suitable for the implementation; no inverse transform is required in the feedback loop.

Our interest in Structure II stems from the fact that many prediction algorithms work in the space domain. Examples are the intraframe coders of Ericson and Forchheimer [18] and Wilson *et al.* [19], [20] and the motion compensated interframe coder of Jain and Jain [12]. Ericson and Forchheimer use a fixed predictor which utilizes five elements in the previous line,

$$B=\begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & & & & & \\ & \cdot & \cdot & \cdot & \cdot & \cdot & & & 0 & \\ & & \cdot & \cdot & \cdot & \cdot & \cdot & & & \\ & & & b_1 & b_2 & b_3 & b_4 & b_5 & & \\ & & & & b_1 & b_2 & b_3 & b_4 & b_5 & \\ & & & & & b_1 & b_2 & b_3 & b_4 & b_5 \\ & & & & & & \cdot & \cdot & \cdot & \cdot \\ & 0 & & & & & & \cdot & \cdot & \cdot \\ & & & & & & & & \cdot & \cdot \end{bmatrix}. \tag{19}$$

The prediction is a low-pass filtered version of the previous reconstructed line, and the prediction error is transformed by a 16-point DCT. The "low-pass filtered previous line" gave a lower variance in the high frequency coefficients than straightforward previous line prediction. A related observation was made by Clarke [22] in a study of Habibi's structure for intraframe hybrid coding. He found that previous line prediction is only suitable for the low-order coefficients [see also (11)].

A hybrid coder with adaptive prediction is shown in Fig. 3. The edge-following adaptive predictor of Wilson *et al.* [19], [20] and the motion compensation scheme by Jain and Jain [12] work in the space domain, which makes Structure II the most suitable. Structure I would give the predictor (17), i.e., the feedback loop would contain both an inverse and a forward transformation. Jain and Jain used a matching algorithm to find the displacement estimates; the displacement vectors for blocks of pels were transmitted as side information.

Pel recursive motion estimation that does not require the transmission of displacement vectors is not applicable to hybrid coding; the previous line and pel are generally not available in the receiver, due to the blockwise processing in the transform coder. A recursive algorithm that works in the transform domain—coefficient recursive motion estimation—was presented by Stuller and Netravali [23] and applied to hybrid coding [11].

Structure II also allows an interesting implementation of a nonlinear recursive filter. In predictive interframe coding, the structure outlined in Fig. 4 has drawn a lot of attention [24]. Before quantization, the prediction error is modified by a function $g(e)$. If $g(e)$ is a linear function,

$$g(e)=\alpha e, \qquad 0<\alpha<1, \tag{20}$$

then the whole system will act as a filter. If straight frame difference prediction is used, we obtain a temporal first-order low-pass recursive filter, which reduces noise in the input signal. As an undesirable side effect we get blurring of moving objects. To overcome this effect, a nonlinear characteristic was introduced by Ishiguro *et al.* [25]. The nonlinear function shown in Fig. 5 only affects small prediction errors. Thus, noise will be attenuated, but high-contrast moving edges will not be affected. The entropy of the prediction error is reduced while maintaining picture quality. Actually, the picture quality could be improved by the noise reduction. If motion-compensated prediction is used in the system, we get a motion-compensated temporal filter which has even fewer problems with blurring of moving edges. The nonlinear pointwise mapping in Fig. 5 can easily be included in Structure II, and temporal filtering will be performed without the cost of an extra frame memory.

An important theoretical result becomes obvious in Structure II. The optimum transform in a hybrid coder is the Karhunen–Loève transform (KLT) for the prediction error [18]. For typical images, this is not the same as the KLT for the original image. They would be the same if the image covariance were separable; however, this is not the case for most images.

C. Hybrid Coding for Line Interlaced TV

Line interlaced TV signals could be handled in a number of ways. Netravali and Stuller [11] and Kamangar and Rao [15] take a two-dimensional transform with all pels in the transform block from the same field. In the DPCM part, the differences between the actual transform coefficients and the coefficients from the same block in the corresponding field one frame back are coded. Hence, the TV signal is treated as one sequence of odd fields and one sequence of even fields which are coded independently.

In a later paper, Kamangar and Rao use the corresponding

Exhibit J Page 196

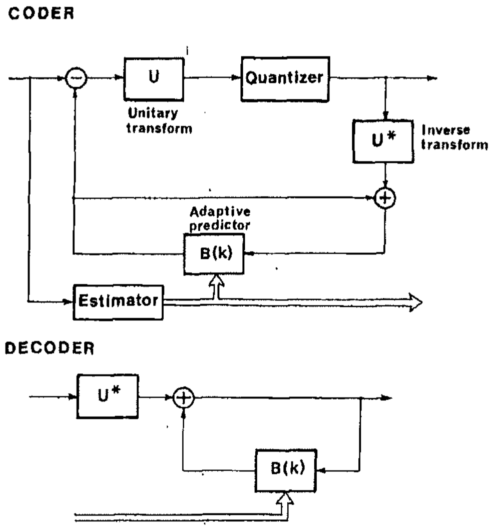


Fig. 3. Hybrid coder with adaptive space domain prediction.

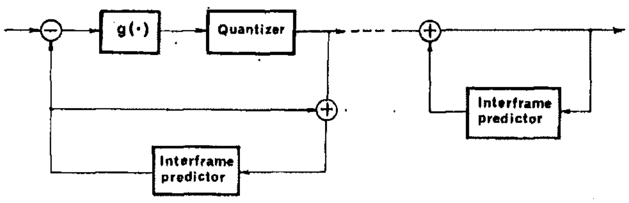


Fig. 4. Predictive system with temporal filtering.

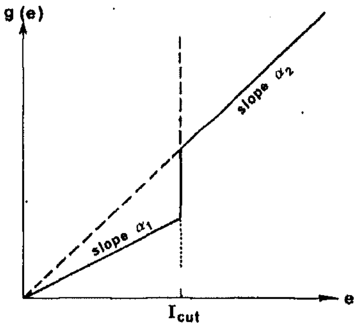


Fig. 5. Nonlinear characteristic (from [24]).

transform block in the previous field [16]. This decreases the memory requirement to one field only, but nonzero prediction errors will be obtained also in nonmoving areas of the picture, as the blocks in the previous field are displaced one line up or down.

Another possibility is to form blocks with pels from both fields and then apply DPCM between frames. The last choice demands the largest memory; the first field of a frame must be stored before it is possible to form the transform blocks.

Structure II offers other possibilities, as the prediction has nothing to do with the transform blocks. In Section IV-B we suggest a simple adaptive space domain predictor: previous frame in unchanged areas and previous field prediction in changed areas. Experiments by Haskell indicate that previous field prediction is advantageous in moving areas [26].

D. Comparison of Structures I and II

We close the section by indicating when to use space domain and when to use transform domain prediction. Transform domain prediction (Structure I) is suitable when the sequences of transform coefficients have different correlation properties, which implies that the prediction coefficients $a_1, \cdots, a_N$ in (10) should not all be equal. It is also suitable when the predictor contains frequency domain processing as in (11). In the special case $A = \alpha I$, Structure I is normally chosen to avoid an inverse transform in the feedback loop.

Structure II should be used when the predictor is applied in the space domain, which is the case for most motion compensation and other adaptive prediction schemes as well as several fixed predictors, i.e., (19). Motion compensation is more easily performed in Structure II, as the displacement must be applied in the space domain. Hence, in Structure I the transform of the displaced block must be found via the space domain.

Structure II also allows the inclusion of a nonlinear temporal filter by a pointwise mapping of the prediction error before transformation; see Figs. 4 and 5. It is also the best structure when some other space domain processing is made, such as the segmentation described by Telese and Zarone [21].

There are other situations when the transform blocks complicate the prediction, as with line interlacing. Structure II also allows staggered transform blocks to diminish the visibility of block borders.

III. Linear Space Domain Prediction

In this section we motivate the use of least mean square (LMS) predictors in a hybrid coder with space domain prediction. First, the usual LMS predictor is derived in matrix notation, then it is shown that the same predictor minimizes the variance for every component after an arbitrary linear transformation; hence, it is also optimum in a hybrid predictive transform coder.

A. LMS Prediction

The predictor in a predictive coder is usually designed with the least mean square criterion. If we perform the analysis for an N-dimensional vector source where M previously transmitted vectors are utilized, we get a predictor of the form

$$\tilde{f}(k) = B(k)g(k) \tag{21}$$

where $g(k)$ is a column vector of dimension MN containing the M previously transmitted vectors,

$$g(k)^T \triangleq (\hat{f}(k-1)^T \cdots \hat{f}(k-M)^T). \tag{22}$$

Equation (21) is a direct generalization of the first-order predictor (14). We get the prediction error vector $e(k)$,

$$e(k) = f(k) - B(k)g(k). \tag{23}$$

The predictor matrix $B(k)$ consists of N vectors $b_i(k)$,

$$B(k) = (b_1(k) \cdots b_N(k))^T. \tag{24}$$

The vector $b_i(k)$ contains the MN predictor coefficients for component i,

$$e_i(k) = f_i(k) - b_i(k)^T g(k), \quad i = 1, \cdots, N. \tag{25}$$

The predictor $B(k)$ shall be chosen such that $E\{e_i(k)^2\}$ is minimized for all components ($i = 1, \cdots, N$). This means that the diagonal elements of the prediction error covariance matrix R_e,

$$R_e(k) \triangleq E\{e(k)e(k)^T\} \tag{26}$$

shall be minimized. Each component i is minimized independently from the other components by a proper choice of the predictor coefficient vector $b_i(k)$. Hence, an equivalent formulation is to minimize the sum of the diagonal elements, i.e., the trace of R_e. The optimal predictor $B(k)$ is found by

Exhibit J Page 197

taking the matrix gradient and equaling it to zero:

$$\text{grad}_{B(k)} \text{ tr } R_e(k) = 0. \tag{27}$$

Using the linearity of the trace operator and the fact that tr A^T = tr A, we obtain

$$\begin{aligned}\text{tr } R_e(k) &= \text{tr } E\{e(k)e(k)^T\} \\ &= \text{tr } E\{[f(k) - B(k)g(k)][f(k)^T - g(k)^T B(k)^T]\} \\ &= \text{tr } (R(k, k) - B(k)R_{gf}(k) \\ &\quad - R_{gf}(k)^T B(k)^T + B(k)R_g(k)B(k)^T) \\ &= \text{tr } R(k, k) - 2 \text{ tr } (B(k)R_{gf}(k)) \\ &\quad + \text{tr } (B(k)R_g(k)B(k)^T)\end{aligned} \tag{28}$$

where we have defined

$$\begin{aligned}R(k_1, k_2) &\triangleq E\{f(k_1)f(k_2)^T\} \\ R_g(k) &\triangleq E\{g(k)g(k)^T\} \\ R_{gf}(k) &\triangleq E\{g(k)f(k)^T\}.\end{aligned} \tag{29}$$

We need two rules for the matrix gradient of a trace:

$$\begin{aligned}\text{grad}_B \text{ tr } (BA) &= A^T \\ \text{grad}_B \text{ tr } (BAB^T) &= BA^T + BA.\end{aligned} \tag{30}$$

The rules in (30) are easily verified by direct computation. By using (28) and (30) in (27), we obtain

$$\begin{aligned}\text{grad}_{B(k)} \text{ tr } R_e(k) = &-2R_{gf}(k)^T \\ &+ B(k)(R_g(k) + R_g(k)^T) = 0.\end{aligned} \tag{31}$$

Since R_g is symmetric, we get

$$B(k)R_g(k) = R_{gf}(k)^T. \tag{32}$$

If R_g has an inverse, the LMS predictor $B^+(k)$ is unique:

$$B^+(k) = R_{gf}(k)^T R_g(k)^{-1}. \tag{33}$$

Equation (32) implies that the LMS prediction error $e^+(k)$ is orthogonal to the data used in the predictor,

$$\begin{aligned}e^+(k) &\triangleq f(k) - B^+(k)g(k) \\ &\Rightarrow \\ R_{e^+g}(k) &\triangleq E\{e^+(k)g(k)^T\} \\ &= R_{gf}(k)^T - B^+(k)R_g(k) = 0,\end{aligned} \tag{34}$$

i.e., the orthogonality principle [27].

If the quantization error is neglected, i.e., $\hat{f}(k) \approx f(k)$, R_g and R_{gf} can be written as block matrices with the covariance for the original data as elements. Equations (22) and (29) give

$$R_{gf}(k) \approx \begin{bmatrix} R(k-1, k) \\ R(k-2, k) \\ \vdots \\ R(k-M, k) \end{bmatrix} \tag{35}$$

$$R_g(k) \approx \begin{bmatrix} R(k-1, k-1) & \cdots & R(k-1, k-M) \\ R(k-2, k-1) & \cdots & R(k-2, k-M) \\ \vdots & \vdots\vdots\vdots & \vdots \\ R(k-M, k-1) & \cdots & R(k-M, k-M) \end{bmatrix}. \tag{36}$$

The LMS criterion is used because of its analytic tractability. We obtain (32), a system of linear equations. In a predictive coder, a more desirable criterion could be to minimize the quantization error with a fixed number of quantization levels; if variable length coding is used, the entropy of the quantized prediction error would be the relevant variable. If the distribution shape of the prediction error is not affected by the predictor choice (e.g., a stationary Gaussian input signal will give a Gaussian prediction error), then the LMS predictor will also satisfy the other criteria.

B. Prediction in a Hybrid Coder

In this section we will show that the LMS predictor also minimizes the mean square of the transform coefficients for any linear transformation. We start by studying an arbitrary linear combination of the prediction error components.

Proposition: If β is an arbitrary linear combination of the prediction error components,

$$\beta \triangleq a^T e(k) \tag{37}$$

where $e(k)$ is defined in (23), then the mean square $E\{\beta^2\}$ is minimized by the LMS predictor B^+, which is a solution to (32).

Proof: Let $R_e(k, B)$ denote the prediction error covariance matrix with predictor B. We study the predictor $B = B^+ + D$, which gives

$$\begin{aligned}R_e(k, B) &= R_e(k, B^+ + D) \\ &= E\{[f(k) - (B^+ + D)g(k)][f(k)^T \\ &\quad - g(k)^T(B^{+T} + D^T)]\} \\ &= R_e(k, B^+) - DR_{e^+g}(k)^T - R_{e^+g}(k)D^T \\ &\quad + DR_g(k)D^T \\ &= R_e(k, B^+) + DR_g(k)D^T.\end{aligned} \tag{38}$$

In the last equality we have utilized the orthogonality principle (34). If we use the above expression, we obtain

$$\begin{aligned}E\{\beta^2\} &= E\{(a^T e(k))^2\} = E\{(a^T e(k))(a^T e(k))^T\} \\ &= E\{a^T e(k)e(k)^T a\} = a^T R_e(k, B)a \\ &= a^T R_e(k, B^+)a + a^T DR_g(k)D^T a.\end{aligned} \tag{39}$$

The last term is nonnegative, since the correlation matrix R_g is nonnegative definite. Hence, we obtain

$$E\{\beta^2\} \geq a^T R_e(k, B^+)a. \tag{40}$$

If D is a zero matrix we get equality in (40), which completes the proof.

It follows immediately that the mean square of each transform coefficient $E_i(k)$,

$$\begin{aligned}E(k) = (E_1(k) \cdots E_N(k))^T &= Ue(k) \\ &= (u_1 \cdots u_N)^T e(k)\end{aligned} \tag{41}$$

is minimized by the LMS predictor, since $E_i(k)$ is a linear combination of the prediction error components

$$E_i(k) = u_i^T e(k), \qquad i = 1, \cdots, N. \tag{42}$$

It could also be noted that a weighted squared error is minimized by the LMS predictor. A weighted square error d is defined as

$$d \triangleq E\{e(k)^T Se(k)\} \tag{43}$$

with S any nonnegative definite symmetric N by N matrix. S can be factored into $W^T W$, where W consists of N row vectors w_i^T with N components each,

$$W^T = (w_1 \cdots w_N). \tag{44}$$

Exhibit J Page 198

If we introduce $x(k) = We(k)$,

$$x(k)=(x_1(k) \cdots x_N(k))^T$$
$$x_i(k)=w_i^T e(k), \quad i=1, \cdots, N \tag{45}$$

we obtain

$$d=E\{e(k)^T W^T We(k)\}=E\{x(k)^T x(k)\}$$
$$=\sum_{i=1}^{N} E\{x_i(k)^2\}. \tag{46}$$

Each term in the sum is minimized by the LMS predictor, since $x_i(k)$ is a linear combination of the prediction error according to (45).

IV. Predictor Experiments

The experiments were performed on two 10 s sequences, "Stenger" and "Wendt," which have been used in the European research project COST 211 [28]. Each frame is 248 pels (picture elements) by 288 interlaced lines, i.e., 144 lines per field. Each pel is digitized with 8 bits resolution to the range −128 to +127. The frame rate is 25 Hz. Sequence "Stenger" is a typical videophone scene with moderate movement. Sequence "Wendt" contains very active motion and a highly detailed background. One frame from each sequence is shown in Fig. 6.

A. Fixed Prediction

The same two sequences have been used in experiments with predictors for DPCM coders [29], [30]. In a DPCM coder, the predictor can use previously transmitted pels in the same field as well as pels in previous fields. In a hybrid interframe coder, the actual block in the actual field is not available to the predictor, i.e., the previous line and previous pel cannot normally be used for prediction.

Table I shows the prediction gain for a number of predictors that could be used in a hybrid coder with space domain prediction (Fig. 2). Prediction gain is defined as

$$20 \cdot \log_{10} \frac{\text{signal rms}}{\text{prediction error rms}} \quad [\text{dB}].$$

LMS predictors using a total of 40 pels in four previously transmitted fields were computed from correlation data for the two sequences, to get an upper bound on the prediction gain obtainable with a fixed linear predictor. The prediction gain is approximately 3 dB higher than for pure frame difference prediction; however, the predictor is very dependent on image statistics. The prediction gain is reduced 1–2 dB when the 40th-order predictor designed for sequence "Wendt" is used on sequence "Stenger" and vice versa.

The third-order predictor uses the corresponding pel in the previous frame and the vertical neighbors which are located in the previous field. From Table I it can be seen that the sequence with moderate movement ("Stenger") puts less weight on the vertical neighbors than sequence "Wendt," which contains large moving areas.

Obviously, it is difficult to find a fixed predictor which gives a considerable advantage over frame difference prediction. In addition, the frame difference predictor is the only fixed predictor which gives small prediction errors in unchanged areas. The histogram of the prediction error is highly peaked, which means that the entropy is low. It was found by Brusewitz [30] that frame difference prediction gave the lowest differential entropy of all fixed predictors. The results are not directly applicable to hybrid coding, as a unitary transform is applied before quantization; however, if variable length coding of the transform coefficients is used, very few bits need to be spent on transform blocks containing background only.

(a)

(b)

Fig. 6. (a) Sequence "Stenger." (b) Sequence "Wendt."

B. Switched Prediction

It is clear that frame difference prediction should be used in background areas. In this section we propose an adaptive predictor which switches between frame difference prediction and a predictor more suitable for moving areas.

Correlations were estimated in the moving areas of the sequences using a motion detector as described in [29]. The first 64 frames from each sequence were used.

Table II shows prediction gain in moving areas for different predictors. It is found that the average of the horizontal and vertical neighbors in the previous field (predictor 2) gives 3 dB better prediction gain compared to frame difference prediction. Higher order LMS predictors were synthesized to obtain a bound on the performance achievable with a linear predictor. An 11th-order predictor using 6 pels in the previous field and 5 pels in the previous frame achieved only 0.5–1 dB higher prediction gain (predictor 3). Predictor 4 uses 18 pels located in four previous fields, but less than 0.1 dB is gained by using pels more than one frame back.

A candidate for further study is a predictor which switches between previous frame and previous field prediction, i.e., between predictors 1 and 2 in Table II. In each block of $K_{hor} \times K_{vert}$ pels, predictor 1 or 2 will be chosen according to some

Exhibit J Page 199

TABLE I
PREDICTION GAIN FOR FIXED LINEAR PREDICTORS

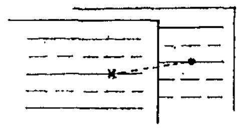
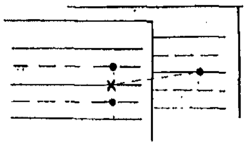
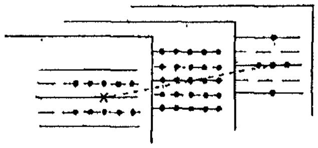
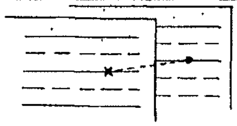
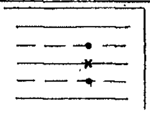
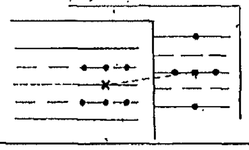
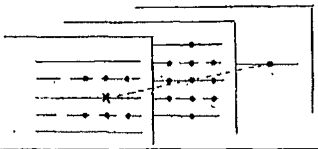
No.	Pixel configuration	Predictor order	Predictor coefficients	Prediction gain (dB) Seq. STENGER	Seq. WENDT
1.		1	Frame difference prediction X'	14.0	9.3
2.		3	$0.72 X' + 0.28 \frac{E+F}{2}$ (Matched to Seq. STENGER)	14.7	10.3
			$0.32 X' + 0.68 \frac{E+F}{2}$ (Matched to Seq. WENDT)	13.3	11.1
3.		40	Matched to Seq. STENGER	17.0	11.8
			Matched to Seq. WENDT	14.8	12.6

TABLE II
GAIN IN CHANGED AREAS FOR FIXED LINEAR PREDICTORS

No.	Pixel configuration	Predictor order	Predictor coefficients	Prediction gain in changed areas (dB) Seq. STENGER	Seq. WENDT
1.		1	X' (Frame difference prediction)	8.2	5.8
2.		2	$\frac{E+F}{2}$ (Previous field prediction)	11.3	9.0
3.		11	Matched to changed areas in Seq. STENGER	11.9	9.9
			Matched to changed areas in Seq. WENDT	11.6	10.1
4.		18	Matched to changed areas in Seq. STENGER	12.0	9.9
			Matched to changed areas in Seq. WENDT	11.7	10.2

criterion, e.g., mean square prediction error. Predictor 1 will mainly be used in unchanged areas and predictor 2 in changed areas. Side information on the predictor choice is sent to the receiver.

C. Motion Compensation

Motion compensation is a more complicated adaptive prediction scheme. The displacement between two pictures is estimated. The predictor is taken to be the previous frame displaced according to a displacement estimate. For comparison to simpler prediction algorithms, we implemented the motion compensation algorithm proposed by Jain and Jain [12], which is a matching algorithm. Numerous other motion compensation algorithms are described in [31].

The algorithm described in [12] performs motion estimation on rectangular blocks, and the estimate is transmitted for each block of $K_{hor} \times K_{vert}$ pels. The estimate is a displacement in integer pels horizontally and vertically, which is found with a logarithmic search algorithm.

We applied the algorithm to the previous field and to the second previous field, which is situated one frame back. A search was performed in each field, and two displacement

Exhibit J Page 200

estimates were obtained. Then the field giving smallest squared error was picked. The maximum allowed displacement was 7 pels horizontally and vertically.

Our simulations showed that it was important to utilize the previous field. More than 60 percent of the blocks with nonzero displacement used the previous field. When a search was performed only in the field situated one frame back, the prediction gain dropped about 2 dB. When a search was performed in the previous field only, and then compared with zero displacement, the loss in prediction gain was only about 0.5 dB. These results indicate that the search could be limited to the previous field with negligible loss compared to a search in two consecutive fields.

Many motion estimation algorithms give a displacement estimate with an accuracy of fractional pels. Then the prediction is formed by linear interpolation in the previous field (or frame). We made a straightforward extension of the algorithm by Jain and Jain [12] and searched for the displacement with an accuracy of 1/8 pel. The matching algorithm by Jain and Jain was kept because of its robustness; the choice was not governed by computational considerations. We found that the search could be limited to motion-compensated previous field and nondisplaced previous frame prediction also in this case, with a loss of around 0.5 dB compared to motion estimation in two previous fields.

D. Simulations

Four predictors were simulated on the two scenes:

- PFra: Previous frame prediction.
- Sw: Switched prediction: previous frame or previous field (average of vertical neighbors). The predictor giving the least squared prediction error is used in each block of $K_{hor} \times K_{vert}$ pels.
- MC-Int: Motion compensation. An integer displacement is estimated for each $K_{hor} \times K_{vert}$ block with the algorithm by Jain and Jain [12]. The two previous fields are searched for a displacement estimate.
- MC-Frac: Motion compensation with fractional displacement estimate. Same as MC-Int, but with a displacement accuracy of 1/8 pel.

To gather statistics, the predictors were simulated in a system without quantization. Prediction gain for the first 16 frames from scenes "Wendt" and "Stenger" is shown in Fig. 7 for different block sizes. Motion compensation with fractional displacement gives the highest prediction gain. It could be seen that the switched previous field/frame predictor is not very sensitive to block size, while the motion compensation algorithms give significantly better performance on small blocks. The block size has to be chosen as a compromise between performance and the amount of side information that has to be sent.

Examples of the prediction error are shown in Fig. 8 for one field. The adaptive predictors used a block size of 16 × 16 pels. The large block size causes problems at the borders between moving area and background; the displacement estimate in a border block is a compromise between the moving area and the nonmoving background, and it will generally be wrong in the whole block, as can be seen in Fig. 8(c) and (d). Some of the background has been displaced, which causes large prediction errors.

The next experiment compares the performance of a hybrid coder and a DPCM coder, i.e., the gain of applying a transform before quantization compared to straightforward quantization of the prediction error. Assuming a logarithmic bit assignment [32] for the transform coefficients in the hybrid

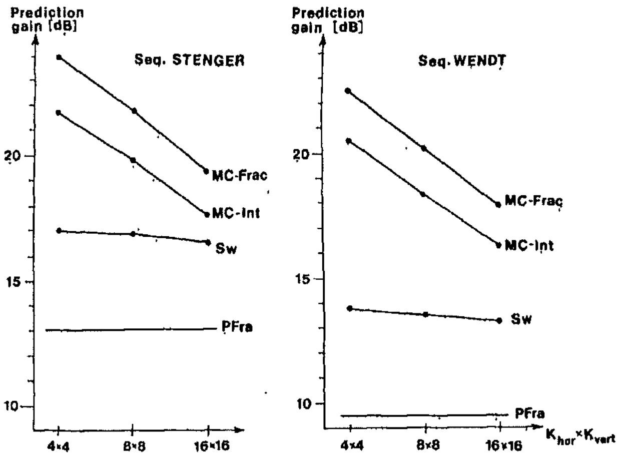


Fig. 7. Prediction gain for adaptive predictors as a function of block size. Simulations on 16 frames from sequences (a) "Stenger" and (b) "Wendt."

coder, we get the bit rate per pel

$$R_h = \frac{1}{N} k \sum_{j=1}^{N} \log_2 \frac{\sigma_j^2}{\theta} \tag{47}$$

where σ_j^2 is the variance of the jth transform coefficient, and θ is a constant that determines the distortion.

In a DPCM coder the prediction error would be coded with

$$R_p = k \log_2 \frac{\sigma^2}{\theta} \quad \text{bits/pel.} \tag{48}$$

The prediction error variance σ^2 in the space domain is related to the variances in the transform domain through Parseval's relation,

$$\sigma^2 = \frac{1}{N} \sum_{j=1}^{N} \sigma_j^2. \tag{49}$$

The prediction errors from the predictors PFra, Sw, MC-Int, and MC-Frac were transformed by a 16 × 16 normalized discrete cosine transform (DCT) and the coefficient variances were computed for the first 64 frames of sequence "Wendt." R_h and R_p according to (47)–(49) are plotted in Fig. 9 with $k = 0.5$ and $\theta = 1$. The values for the original pictures are also included to give a comparison with intraframe transform coding. The value $k = 0.5$ is the rate distortion bound for a stationary Gaussian source, and $\theta = 1$ corresponds to 48 dB SDR.[1] This is a very high quality, which explains the high bit rate estimates. The distance between the two curves indicates the compression that can be expected from the transform in the hybrid coder. Clearly, the use of a more sophisticated predictor is less advantageous in a hybrid coder than in a predictive coder. The reason is that the adaptive predictors give a flatter spectrum. This is illustrated in Fig. 10, where standard deviation estimates for transform coefficients along the main diagonal are plotted.

V. Simulation of an Adaptive Hybrid Coder

A. Transform Coder

In the previous section the transformed prediction error was never quantized. A complete system contains an intraframe

[1] The signal-to-distortion ratio (SDR) is defined as

$$\text{SDR} = 20 \cdot \log_{10} \frac{\text{Peak-to-peak signal value}}{\text{RMS coding error}} \quad \text{[dB]}.$$

The peak-to-peak value is 255 in our case.

Exhibit J Page 301

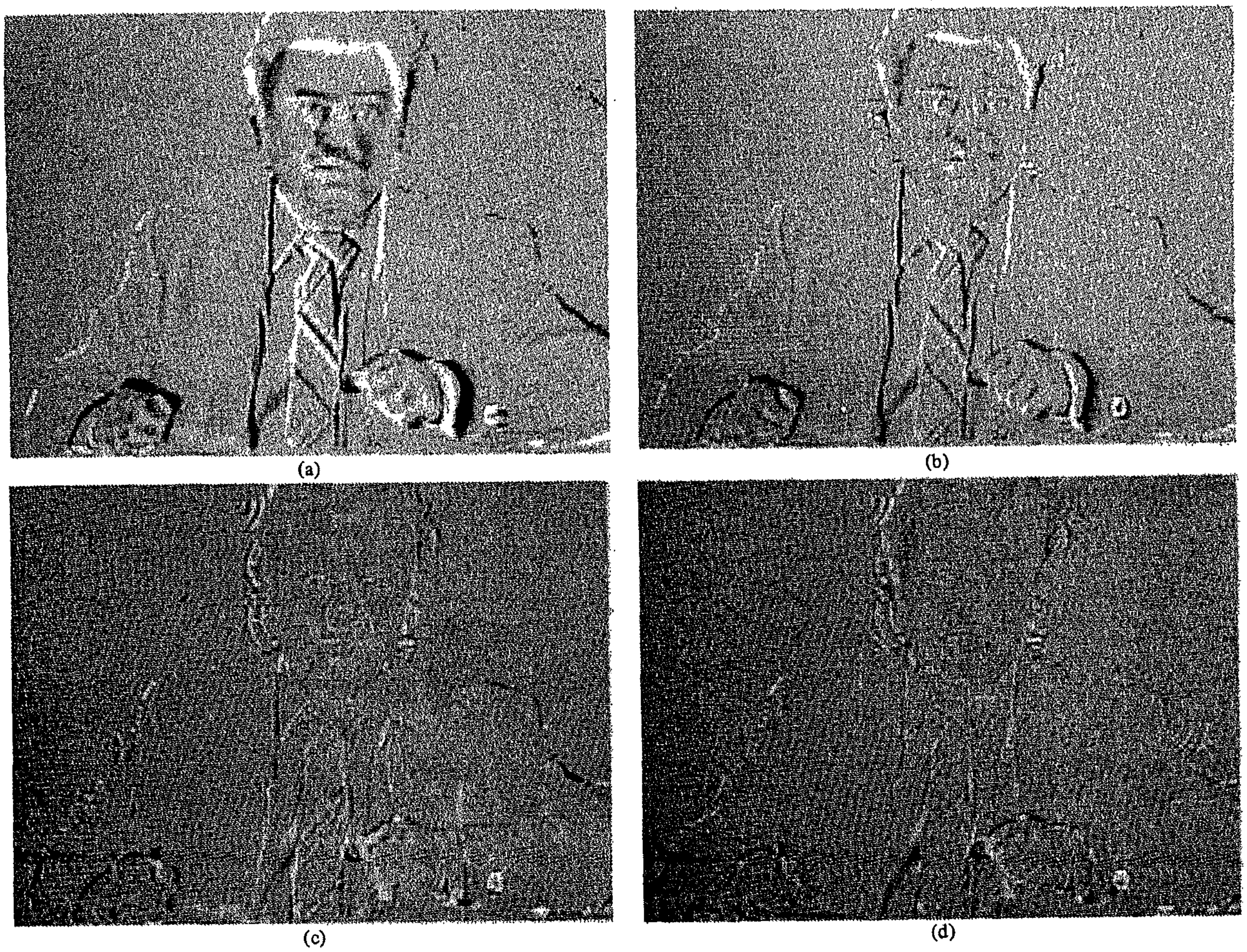


Fig. 8. Prediction error with different predictors. (a) PFra: previous frame. (b) Sw: switched previous field/previous frame. (c) MC-Int: motion compensation, integer pel displacements. (d) MC-Frac: motion compensation, 1/8 pel accuracy.

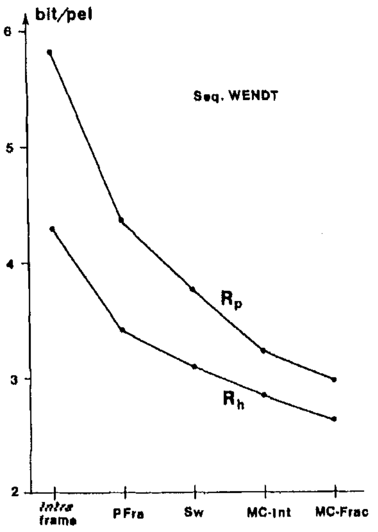

bit/pel
6
5
4
3
2
Seq. WENDT
R_p
R_h
Intra frame
PFra
Sw
MC-Int
MC-Frac

Fig. 9. Bit rate estimates with different predictors for purely predictive coder [R_p, defined in (48)] and hybrid coder [R_h, defined in (47)]. Distortion parameter $\theta = 1$.

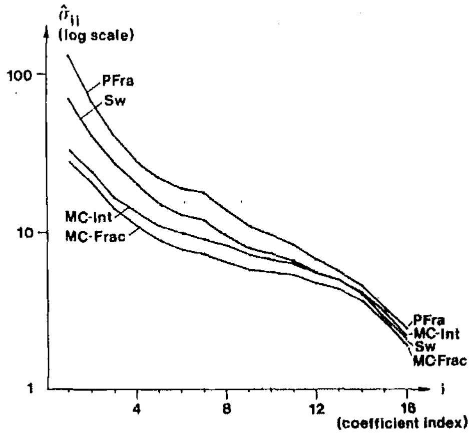

$\hat{\sigma}_{ii}$
(log scale)
100
10
1
PFra
Sw
MC-Int
MC-Frac
PFra
MC-Int
Sw
MCFrac
4
8
12
16
i
(coefficient index)

Fig. 10. Standard deviation for a normalized 16 × 16 DCT of the prediction error. Computed from the first 64 frames of sequence "Wendt." Plot contains estimates for coefficients (i, i), $i = 1, \cdots, 16$, where (1, 1) is the DC coefficient.

Exhibit J Page 202

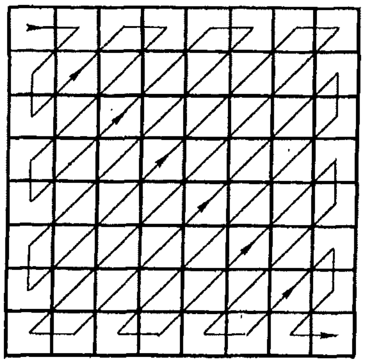


Fig. 11. Diagonal scanning of transform coefficients.

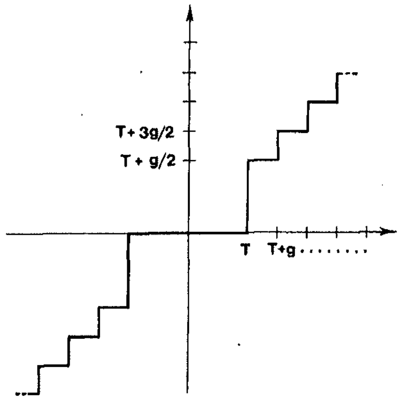


Fig. 12. Threshold coder quantization characteristic.

transform coder which operates on difference pictures; see Fig. 2. Typical examples of the input to the transform coder are given in Fig. 8. The varying statistics in background and moving area indicate that an adaptive coder should be advantageous. We have used a coder similar to the scene adaptive coder of Chen and Pratt [33] for the intraframe transform coding module outlined in Fig. 2. The scene adaptive coder has shown good performance as an intraframe coder; here we test it in a new environment.

A 16 × 16 discrete cosine transform is used. The transform coefficients are quantized by a threshold coder, i.e., all coefficients with a magnitude above a threshold are addressed and quantized. To make the addressing efficient, each block of transform coefficients is scanned in a diagonal fashion; see Fig. 11. Coefficients less than the threshold are set to zero and runlength coded. Coefficients above threshold are quantized with a uniform quantizer; see Fig. 12. The runlengths and quantized values are coded with separate Huffman code tables. The output bit rate is variable but can be controlled by adjusting the threshold and the quantization step, T and g, respectively, in Fig. 12. A threshold setting of $T = 1.5g$ was found to work well over a wide range of bit rates.

B. Results

Simulations were performed with fieldwise coding of the sequences. The four different predictors listed in Section IV-D were used. Results are summarized in Fig. 13. The first 64 frames from each sequence were coded with three different quantization steps: $g = 5$, 10, and 20. A threshold $T = 1.5g$ was used. The three quantizers gave around 40, 35, and 30 dB SDR, respectively, on sequence "Stenger." The SDR was slightly lower for sequence "Wendt." The coding was also made on full frames with frame difference prediction and

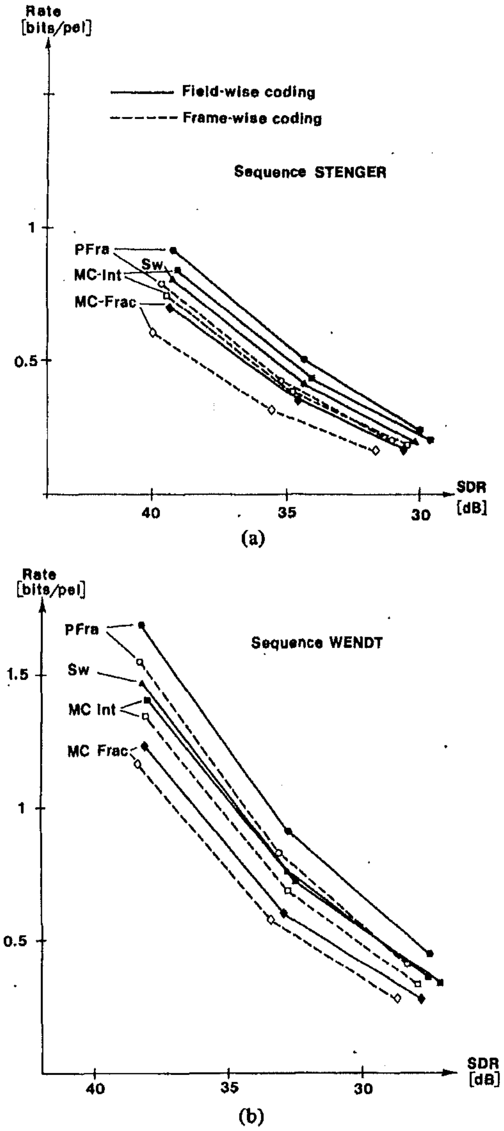


Fig. 13. Bit rate versus distortion for hybrid coding of first 64 frames from sequences (a) "Stenger" and (b) "Wendt."

motion compensation. Full frame coding performs better than fieldwise coding. The reason is that the vertical sampling step is halved, which increases vertical correlation and the performance of the transform coder. The motion compensation algorithms give 1 or 2 dB less prediction gain compared to fieldwise processing, but this effect is obviously weaker than the gain in the transform coder.

Integer motion compensation is comparable to the switched predictor, giving 15–20 percent bit rate saving. The motion compensator with fractional pel accuracy is clearly the best, with 25–35 percent less bit rate than previous frame prediction. Another attractive property of motion compensation is that the bit rate varies more smoothly, as can be seen in Fig. 14.

An informal subjective evaluation was performed on sequence "Wendt" coded with quantization step $g = 10$ and threshold $T = 15$, which gave approximately 33 dB SDR. Generally, all systems gave good quality in the background. The distortions were perceived in moving areas and in background close to moving areas. Previous frame prediction

Exhibit J Page 203

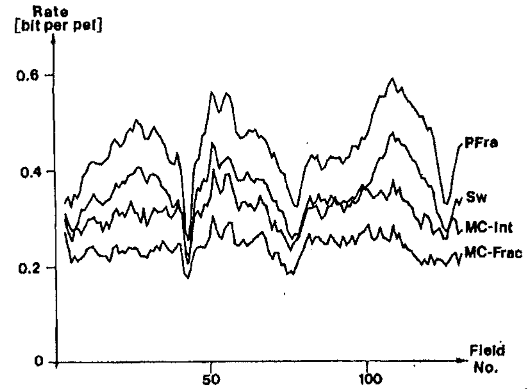


Fig. 14. Bit rate for each field with four different predictors. Sequence "Wendt," 64 frames, $g = 20$, $T = 30$.

(PFra) made the worst subjective impression, and motion compensation with fractional displacement (MC-Frac) looked best. The distortion with PFra gave a "dirty window" effect, i.e., a nonmoving noise pattern was superimposed on the moving parts of the picture. The switched field/frame predictor (Sw) gave a similar effect, although the dirty window contained less high spatial frequency noise in the vertical direction. The reduction of vertical HF noise in moving areas is due to the predictor, which uses an average of the vertical neighbors in the previous field. Hence, the prediction is a filtered version of the previous field.

Motion compensation with integer pel displacement (MC-Int) contained approximately the same amount of noise as PFra, but it was less annoying. It was moving together with the picture details, which made it less visible than a nonmoving "dirty window." MC-Frac looked even better. The noise was reduced in most parts of the moving area, although it could still be perceived near borders between moving area and background. Obviously, the interpolation performed by MC-Frac to get a noninteger displacement gives a filtering effect which is very advantageous; quantization errors in the previous frame are reduced.

Full frame coding gave the same overall impression as fieldwise coding. We observed a difference with motion compensation: the noise near borders between moving area and background did not extend as far from the borders as with fieldwise coding. It is explained by the fact that a 16×16 block in a field has twice the height of a 16×16 block in a frame composed of two interlaced fields.

Concluding, we find that motion compensation should be performed with fractional pel accuracy to be effective. The reduced dirty window effect makes motion compensation even more advantageous than indicated by the rate versus SDR figures.

Framewise processing is also of interest, with a bit saving of 5–20 percent compared to fieldwise coding. The drawback is that an extra field buffer is needed. An extra delay is also introduced by the coder. Field coding with switched prediction is an attractive alternative to framewise coding with frame difference prediction.

Appendix

The equivalence between Structures I and II, transform domain prediction (Fig. 1) and space domain prediction (Fig. 2), respectively, can be shown with elementary block manipulations. If we start with Fig. 1 and insert cascaded inverse and forward transforms as outlined in Fig. 15, we have not changed the system. As U is a linear transformation, the order between transform and addition/subtraction can be changed. This has been done in Fig. 16. A comparison with Fig. 2 shows that the two structures are equivalent if we keep the same quantizer, and if the space domain predictor B in Fig. 2 is chosen to be

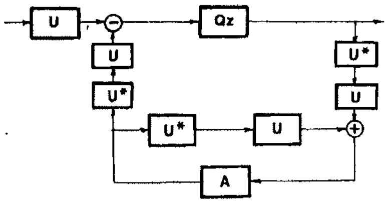


Fig. 15.

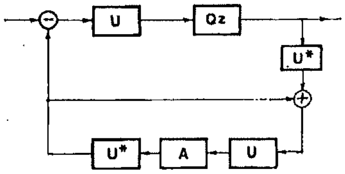


Fig. 16.

$$B = UAU^*.$$

Acknowledgment

Dr. R. Forchheimer has influenced the work to a large extent by several stimulating discussions. I would also like to thank him and his colleagues, S. Cronström and T. Kronander at Linköping University, for their kind help with D-to-A conversion of the processed picture sequences. The digitized picture sequences were provided by Dr. L. Stenger at the Forschungsinstitut der DBP, Darmstadt, West Germany.

I am also grateful for the encouragement of Prof. L. H. Zetterberg, and for his valuable comments on the report that this paper is based upon.

References

[1] A. N. Netravali and J. O. Limb, "Picture coding: A review," *Proc. IEEE*, vol. 68, pp. 366–406, Mar. 1980.
[2] A. K. Jain, "Image data compression: A review," *Proc. IEEE*, vol. 69, pp. 349–389, Mar. 1981.
[3] A. Habibi, "Hybrid coding of pictorial data," *IEEE Trans. Commun.*, vol. COM-22, pp. 614–624, May 1974.
[4] ——, "An adaptive strategy for hybrid image coding," *IEEE Trans. Commun.*, vol. COM-29, pp. 1736–1740, Dec. 1981.
[5] A. Ploysongsang and K. R. Rao, "DCT/DPCM processing of NTSC composite video signal," *IEEE Trans. Commun.*, vol. COM-30, pp. 541–549, Mar. 1982.
[6] L. Chiariglione, M. Guglielmo, and A. Sciarappa, "Transform coding of PAL composite signals," *CSELT Rapporti tecnici*, vol. VII, pp. 109–118, June 1979.
[7] J. A. Roese, A. Habibi, W. K. Pratt, and G. S. Robinson, "Interframe transform coding and predictive coding methods," in *Proc. Int. Conf. Commun.*, 1975, vol. 2, pp. 23-17–23-21.
[8] J. A. Roese, "Interframe coding of digital images using transform and hybrid transform/predictive techniques," Image Processing Inst., Univ. Southern California, Los Angeles, USCIPI Rep. 700, June 1976.
[9] J. A. Roese, W. K. Pratt, and G. S. Robinson, "Interframe cosine transform image coding," *IEEE Trans. Commun.*, vol. COM-25, pp. 1329–1339, Nov. 1977.
[10] J. A. Roese, "Hybrid transform/predictive image coding," in *Image Transmission Techniques*, W. K. Pratt, Ed. New York: Academic, 1979.
[11] A. N. Netravali and J. A. Stuller, "Motion-compensated transform coding," *Bell Syst. Tech. J.*, vol. 58, pp. 1703–1718, Sept. 1979.
[12] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. COM-29, pp. 1799–1808, Dec. 1981.

Exhibit J Page 204

[13] F. May, "Hybrid coding of picture sequences for transmission over narrowband mobile radio channels," in *Proc. Euro. Signal Processing Conf. (EUSIPCO)*, Sept. 1980, pp. 283-288.
[14] W. A. Pearlman and P. Jakatdar, "Hybrid DFT/DPCM interframe image quantization," in *Proc. ICASSP*, 1981, vol. 3, pp. 1121-1124.
[15] F. A. Kamangar and K. R. Rao, "Interframe hybrid coding of NTSC component video signal," in *Proc. Nat. Telecommun. Conf.*, 1979, pp. 53.2.1-53.2.5.
[16] ——, "Interfield hybrid coding of component color television signals," *IEEE Trans. Commun.*, vol. COM-29, pp. 1740-1753, Dec. 1981.
[17] R. Forchheimer, "Differential transform coding—A new hybrid coding scheme," presented at Picture Coding Symp., Montreal, P.Q. Canada, June 3-5, 1981.
[18] T. Ericson and R. Forchheimer, "Linear transformations for optimal quantization of vector processes with an application to picture coding," Dep. Elec. Eng., Linköping Univ., Linköping, Sweden, Internal Rep. LiTH-ISY-I-0449, 1981.
[19] R. Wilson, H. E. Knutsson, and G. H. Granlund, "Image coding using a predictor controlled by image content," in *Proc. ICASSP*, 1982, vol. 1, pp. 432-435.
[20] ——, "Anisotropic non-stationary image estimation and its applications: Part II—Predictive image coding," *IEEE Trans. Commun.*, vol. COM-31, pp. 398-406, Mar. 1983.
[21] O. Telese and G. Zarone, "Video hybrid coding by picture domain segmentation," in *Proc. 8eme Colloque sur le Traitement du Signal et Ses Applications*, Nice, France, June 1-5, 1981, pp. 747-752.
[22] R. J. Clarke, "Hybrid intraframe transform coding of image data," *IEE Proc.*, Part F, vol. 131, pp. 2-6, Feb. 1984.
[23] J. A. Stuller and A. N. Netravali, "Transform domain motion estimation," *Bell Syst. Tech. J.*, vol. 58, pp. 1673-1702, Sept. 1979.
[24] E. Dubois, B. Prasada, and M. S. Sabri, "Image sequence coding," in *Image Sequence Analysis*, T. S. Huang, Ed. Berlin, Germany: Springer-Verlag, 1981.
[25] T. Ishiguro, K. Iinuma, Y. Iijima, T. Koga, S. Azami, and T. Mune, "Composite interframe coding of NTSC color television signals," in *Nat. Telecommun. Conf. Rec.*, 1976, vol. 1, pp. 6.4-1-6.4-5.
[26] B. G. Haskell, "Entropy measurements for nonadaptive and adaptive frame-to-frame, linear-predictive coding of video-telephone signals," *Bell Syst. Tech. J.*, vol. 54, pp. 1155-1174, July-Aug. 1975.
[27] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, 1965.
[28] J. E. Thompson, "European collaboration on picture coding research for 2 Mbit/s transmission," *IEEE Trans. Commun.*, vol. COM-29, pp. 2003-2004, Dec. 1981.
[29] S. Ericsson, "Fixed and adaptive predictors for picture coding," Telecommun. Theory, Royal Inst. Technol., Stockholm, Sweden, Rep. TRITA-TTT-8103, Apr. 1981.
[30] H. Brusewitz, "Prediction in DPCM coding of moving images," Telecommun. Theory, Royal Inst. Technol., Stockholm, Sweden, Rep. TRITA-TTT-8203, Sept. 1982.
[31] T. S. Huang, Ed., *Image Sequence and Dynamic Scene Analysis* (NATO ASI Series F:2). Berlin, Germany: Springer-Verlag, 1983.
[32] A. Segall, "Bit allocation and encoding for vector sources," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 162-169, Mar. 1976.
[33] W. -H. Chen and W. K. Pratt, "Scene adaptive coder," *IEEE Trans. Commun.*, vol. COM-32, pp. 225-232, Mar. 1984.

★

Staffan Ericsson was born in Finspong, Sweden, on July 14, 1954. He received the M.S. degree in electrical engineering in 1978 and the Ph.D. degree in 1984, both from the Royal Institute of Technology, Stockholm, Sweden.

From 1984 to 1985 he was a Visiting Research Associate at INRS-Telecommunications, Montreal, P.Q., Canada. In 1985 he joined PicTel Corporation, Peabody, MA. His research interests include image data compression and processing.

Exhibit J Page 205